HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

11-29-05

AF/2192

PATENT APPLICATION

ATTORNEY DOCKET NO. 10013502-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Joubert Berger et al.

Confirmation No.: 2270

Application No.: 09/896,351

Examiner: C. O. Kendall

Filing Date: June 29, 2001

Group Art Unit: 2192

Title: SYSTEM AND METHOD FOR TRANSFORMING OPERATING SYSTEM AUDIT DATA TO A DESIRED FORMAT

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on **Sept. 28, 2005**.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

( ) (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d) for the total number of months checked below:

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

( ) The extension fee has already been filled in this application.

(X) (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account **08-2025** the sum of **$500.00**. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV 568260183US addressed to: Commissioner for Patents, Alexandria, VA 22313-1450
Date of Deposit: Nov. 28, 2005

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number_____ on _____

Number of pages:

Typed Name: Joy H. Perigo

Signature: Joy H. Perigo

Respectfully submitted,

Joubert Berger et al.

By _____

Jody C. Bishop

Attorney/Agent for Applicant(s)
Reg. No. **44,034**

Date: **Nov. 28, 2005**

Telephone No.: **(214) 855-8007**

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 10013502-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Joubert Berger et al.

Application No.: 09/896,351 | Confirmation No.: 2270

Filed: June 29, 2001 | Art Unit: 2192

For: SYSTEM AND METHOD FOR TRANSFORMING OPERATING SYSTEM AUDIT DATA TO A DESIRED FORMAT | Examiner: C. O. Kendall

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on September 28, 2005, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I.	Real Party In Interest
II	Related Appeals and Interferences
III.	Status of Claims
IV.	Status of Amendments
V.	Summary of Claimed Subject Matter
VI.	Grounds of Rejection to be Reviewed on Appeal

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 54 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-54
4. Claims allowed: None
5. Claims rejected: 1-54

C. Claims On Appeal

The claims on appeal are claims 1-54

IV. STATUS OF AMENDMENTS

The present application was filed June 29, 2001. Responsive to a first Office Action, mailed January 20, 2004, Applicant submitted a Response on February 18, 2004, which presented an amendment to claim 37. A second, non-final, Office Action was then mailed May 7, 2004, and Applicant submitted a Response on August 6, 2004, which amended claim 28 and added new claims 40-54. A Final Office Action was then mailed January 6, 2005, and in response Applicant filed, on April 6, 2005, a Request for Continued Examination (RCE) with an accompanying amendment, which amended claims 1, 5-7, 14, 15, 20, 21, 23, 26, 27, 31, 36-39, 47, and 54. An Office Action was then mailed June 29, 2005, from which the present appeal was taken. Thus, Applicant did not file an Amendment in response to the June 29, 2005 Office Action, but instead filed a Notice of Appeal which this brief supports. Accordingly, the claims on appeal are those rejected in the Office Action of June 29, 2005, a complete listing of which are provided in Appendix A.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

According to one claimed embodiment, such as that of claim 1, a system comprises an operating system (e.g., operating system 101 of FIGURE 1, and *see* page 5, lines 1-9 of the specification) providing at least one routine capable of being invoked, and the operating system is operable to collect (e.g., via auditing function 201 of FIGURES 2-6, and *see* page 5, lines 1-9 and page 8, line 1 – page 9, line 2 of the specification) raw audit data for invoked operating system routines. The system further comprises data storage (e.g., data storage 202 in FIGURES 2-6, and *see* page 5, lines 1-9 and page 9, lines 3-11 of the specification) having said raw audit data stored thereto. The system further comprises software code (e.g., data transform application 301 of FIGURES 3-6; and *see* page 5, lines 1-9 and page 14, lines 11-23 of the specification) executable by at least one processor to receive the raw audit data and

generate output (e.g., output 303 of FIGURE 3, output 303A of FIGURE 4, output 303B of FIGURE 5, and output 303c of FIGURE 6; and *see* page 5, lines 1-9 and page 14, lines 11-23 of the specification) comprising at least a portion of the raw audit data in a desired format defined by a template (e.g., template 302 of FIGURE 3, template 302A of FIGURE 4, template 302B of FIGURE 5, and template 302c of FIGURE 6; and *see* page 5, lines 1-9 and page 14, line 20 – page 16, line 3 of the specification).

In certain embodiments, for example that of claim 2, the template comprises at least one constant element (*see* e.g., page 20, lines 3-17 of the specification). In certain embodiments, such as that of claim 3, the at least one constant element is included verbatim in the output (*see* e.g., page 20, lines 3-17 of the specification).

In certain embodiments, such as that of claim 5, the template comprises at least one variable element that identifies a particular portion of the raw audit data to be included in the output (*see* e.g., page 20, line 3 – page 22, line 5 of the specification). In certain embodiments, such as that of claim 6, the variable element identifies a location within the output at which the particular portion of the raw audit data is to be arranged (*see* e.g., page 20, line 3 – page 22, line 5 of the specification).

In certain embodiments, for example that of claim 10, the template comprises at least one conditional element (*see* e.g., page 26, line 7 – page 27, line 7 of the specification). In certain embodiments, such as that of claim 11, the at least one conditional element dictates that the output is to have a particular format if a condition is satisfied, otherwise the output is to have a different format (*see* e.g., page 26, line 7 – page 27, line 7 of the specification).

In certain embodiments, such as that of claim 40, the generated output comprises presentation output (e.g., presentation output 304 of FIGURE 3). The presentation output may comprise presentation output to a display or presentation output to a printer, as examples (*see* page 14, line 28 – page 15, line 1 of the specification), such as recited in claim 41. In certain embodiments, such as that of claim 42, the presentation output comprises presentation output by a browser, presentation output by a spreadsheet program, or presentation output by an application program (*see* page 14, line 20 – page 15, line 11 of the specification).

In certain embodiments, such as that of claim 43, the system further comprises a user interface for receiving from a user input defining the template (*see* page 12, lines 3-15 of the specification).

According to another claimed embodiment, such as that of claim 14, a computer program product is provided for generating audit data in a desired format, wherein the audit data relates to execution of a routine. The computer program product comprises a computer-readable storage medium having computer-readable program code embodied in the medium. The computer-readable program code comprises code (e.g., data transform application 301 of FIGURES 3-6; and *see* page 5, lines 10-19 and page 14, lines 11-23 of the specification) executable to access raw audit data stored in a data storage device (e.g., data storage 202 in FIGURES 2-6, and *see* page 5, lines 10-19 and page 9, lines 3-11 of the specification), wherein the raw audit data comprises information relating to execution of at least one invoked routine. The computer-readable program code further comprises code executable to access an audit transformation template (e.g., template 302 of FIGURE 3, template 302A of FIGURE 4, template 302B of FIGURE 5, and template 302c of FIGURE 6; and *see* page 5, lines 10-19 and page 14, line 20 – page 16, line 3 of the specification), and code executable to generate output (e.g., output 303 of FIGURE 3, output 303A of FIGURE 4, output 303B of FIGURE 5, and output 303c of FIGURE 6; and *see* page 5, lines 10-19 and page 14, lines 11-23 of the specification) comprising at least a portion of the raw audit data, wherein the output has a format defined by the audit transformation template.

In certain embodiments, such as that of claim 19, the audit transformation template comprises at least one constant element that is included verbatim in the output (*see* e.g., page 20, lines 3-17 of the specification).

In certain embodiments, such as that of claim 24, the template comprises at least one variable element, wherein the at least one variable element each identify a particular type of audit information to be included in the output (*see* e.g., page 20, line 3 – page 22, line 5 of the specification).

In certain embodiments, such as that of claim 25, the template comprises at least one conditional element, wherein the conditional element dictates that the output is to have a first

format if a condition is satisfied and have a different format if the condition is not satisfied (*see* e.g., page 26, line 7 – page 27, line 7 of the specification).

In certain embodiments, such as that of claim 44, the code executable to generate output comprises code executable to generate presentation output (e.g., presentation output 304 of FIGURE 3). The presentation output may be presentation output to a display or presentation output to a printer, as examples (*see* page 14, line 28 – page 15, line 1 of the specification), such as recited by claim 45. In certain embodiments, such as that of claim 46, the presentation output comprises presentation output by a browser, presentation output by a spreadsheet program, or presentation output by an application program (*see* page 14, line 20 – page 15, line 11 of the specification).

In certain embodiments, such as that of claim 47, the computer-executable code further comprises code executable to receive from a user input defining the audit transformation template (*see e.g.,* page 12, lines 3-15 of the specification).

According to another claimed embodiment, for example as in claim 26, a method of generating an output that includes audit data therein and has a desired format is provided. The method comprises collecting (e.g., via auditing function 201 of FIGURES 2-6, and *see* page 5, line 20 – page 6, line 2 and page 8, line 1 – page 9, line 2 of the specification) raw audit data relating to the execution of one or more invoked routines, and storing the raw audit data to a data storage device (e.g., data storage 202 in FIGURES 2-6, and *see* page 5, line 20 – page 6, line 2 and page 9, lines 3-11 of the specification). The method further comprises accessing the raw audit data, and accessing an audit transformation template (e.g., template 302 of FIGURE 3, template 302A of FIGURE 4, template 302B of FIGURE 5, and template 302c of FIGURE 6; and *see* page 5, line 20 – page 6, line 2 and page 14, line 20 – page 16, line 3 of the specification) that defines a desired format. The method further comprises generating an output (e.g., output 303 of FIGURE 3, output 303A of FIGURE 4, output 303B of FIGURE 5, and output 303c of FIGURE 6; and *see* page 5, line 20 – page 6, line 2 and page 14, lines 11-23 of the specification) that includes at least a portion of the raw audit data, wherein the output comprises the desired format as defined by the audit transformation template.

In certain embodiments, such as that of claim 28, the method further comprises receiving input from a user for creating the audit transformation template (*see e.g.,* page 20, lines 3-4 of the specification).

In certain embodiments, such as that of claim 29, the audit transformation template comprises at least one constant element that is included verbatim in the output (*see* e.g., page 20, lines 3-17 of the specification).

In certain embodiments, such as that of claim 33, the method further comprises presenting the output to a user (e.g., presentation 304 of FIGURE 3).

In certain embodiments, such as that of claim 48, generating an output comprises generating an output presentation (e.g., output presentation 304 of FIGURE 3). The output presentation may be an output presentation to a display or an output presentation to a printer, as examples (*see* page 14, line 28 – page 15, line 1 of the specification), as recited by claim 49. In certain embodiments, such as that of claim 50, the output presentation comprises output presentation by a browser, output presentation by a spreadsheet program, or output presentation by an application program (*see* page 14, line 20 – page 15, line 11 of the specification).

According to another claimed embodiment, for example in claim 37, a library of software functions stored to a computer-readable medium comprises a function executable to access raw audit data collected by an auditing program (e.g., auditing function 201 of FIGURES 2-6, and *see* page 6, lines 3-12 and page 8, line 1 – page 9, line 2 of the specification), wherein the raw audit data comprises information about at least one invoked routine of the operating system. The library further comprises a function executable to access a template (e.g., template 302 of FIGURE 3, template 302A of FIGURE 4, template 302B of FIGURE 5, and template 302c of FIGURE 6; and *see* page 6, lines 3-12 and page 14, line 20 – page 16, line 3 of the specification) defining an output format, and a function executable to generate output (e.g., output 303 of FIGURE 3, output 303A of FIGURE 4, output 303B of FIGURE 5, and output 303c of FIGURE 6; and *see* page 6, lines 3-12 and page 14, lines 11-23 of the specification) comprising at least a portion of the raw audit data, wherein the output has a format defined by the template.

In certain embodiments, such as that of claim 51, the function executable to generate output comprises a function executable to generate output presentation (e.g., output presentation 304 of FIGURE 3). The output presentation may be an output presentation to a display or an output presentation to a printer, as examples (*see* page 14, line 28 – page 15, line 1 of the specification), as recited in claim 52. In certain embodiments, such as that of claim 53, the output presentation comprises output presentation by a browser, output presentation by a spreadsheet program, or output presentation by an application program (*see* page 14, line 20 – page 15, line 11 of the specification).

According to another claimed embodiment, such as that of claim 54, a method of generating an output presentation (e.g., output presentation 304 of FIGURE 3) that includes audit data therein and has a desired format is provided. The method comprises receiving input defining an audit transformation template (e.g., template 302 of FIGURE 3, template 302A of FIGURE 4, template 302B of FIGURE 5, and template 302c of FIGURE 6; and *see* page 14, line 20 – page 16, line 3 of the specification) that defines a desired format for the output presentation. The method further comprises collecting (e.g., via auditing function 201 of FIGURES 2-6, and *see* page 8, line 1 – page 9, line 2 of the specification) raw audit data relating to the execution of one or more invoked routines, and storing the raw audit data to a data storage device (e.g., data storage 202 in FIGURES 2-6, and *see* page 9, lines 3-11 of the specification). The method further comprises accessing the raw audit data, and accessing the audit transformation template that defines a desired format. The method further comprises generating (e.g., via data transform application 301 of FIGURES 3-6) the output presentation (e.g., output presentation 304 of FIGURE 3 and output presentation 304c of FIGURE 6) that includes at least a portion of the raw audit data, wherein the output presentation comprises the desired format as defined by the audit transformation template.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 10, 11, 13-18, 37, 40-47, and 51-54 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,347,374 issued to Drake et al. (hereinafter "*Drake*");

Claims 2-9, 19-36, 38-39, and 48-50 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Drake* in view of U.S. Patent No. 5,920,719 issued to Sutton et al. (hereinafter "*Sutton*"); and

Claim 12 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Drake* in view of U.S. Patent No. 6,253,337 issued to Maloney et al. (hereinafter "*Maloney*").

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. Instead, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

I. Rejections Under 35 U.S.C. § 102(e) over *Drake*

Claims 1, 10, 11, 13-18, 37, 40-47, and 51-54 are rejected under 35 U.S.C. § 102(e) as being anticipated by *Drake*. Appellant respectfully traverses this rejection as provided further below.

To anticipate a claim under 35 U.S.C. § 102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. Appellant respectfully submits that *Drake* fails to teach each and every element of claims 1, 10, 11, 13-18, 37, 40-47, and 51-54.

Independent Claims 1, 14, and 37; and Dependent Claims 13 and 15-18

Drake fails to teach each of the elements of independent claims 1, 14, and 37. For instance, independent claim 1 recites in part "software code executable by at least one processor to receive said raw audit data and generate output comprising at least a portion of said raw audit data in a desired format defined by a template" (emphasis added).

Similarly, independent claim 14 recites in part "code executable to generate output comprising at least a portion of said raw audit data, said output having a format defined by said audit transformation template" (emphasis added).

Independent claim 37 recites in part "function executable to generate output comprising at least a portion of said raw audit data, wherein said output has a format defined by said template" (emphasis added).

Drake fails to teach at least the above elements of these independent claims. As discussed below, *Drake* fails to teach generating an output that comprises raw audit data and has a format defined by a template. *Drake* teaches a system in which raw audit data is captured and is converted into a normalized, standard format. The normalized data, rather than the raw audit data, is stored to a database, and such database may be queried to generate output in accordance with a template. Thus, while *Drake* mentions use of templates for retrieving normalized audit data from the database, *Drake* provides no teaching or suggestion of a template for defining an output comprising raw audit data. Rather, *Drake* transforms its raw audit data to a normalized format, which is stored to a database, and a GUI accesses such normalized data rather than the raw audit data.

Drake at col. 5, lines 21-32 provides an event detection system:

> which can be viewed as a dual three-tiered implementation with a database 12 in the middle. On one side is an audit analysis engine 14, which converts raw audit data into a standardized format, and performs expert system analysis on the data. On the other side is a user interface 16, which consists of management and control functions, and an application user interface that provides data mining tools to the use of the invention referred to herein as the event detection system.

In *Drake*, events are:

> stored in relational database 12 in a normalized format, i.e., standard, that maximizes storage capacity and flexibility. The normalized format also simplifies analysis of events, in that no matter what the audit source 18, the events are represented in a single format. Col. 5, lines 62-67.

External to the database 12, events are passed between processes in a standardized representation referred to as a Virtual Record. The Virtual Record is a standardized flat representation of an event in normalized format. Col. 6, lines 4-8. "A parser 20 performs the audit parsing, and has as its sole function the conversion of raw event records into Virtual Records." Col. 7, lines 37-39.

Drake shows in FIG. 1 that a GUI 16 is operable to interact with CMDS database 12 (to which the audit data is stored in a normalized format). For instance, *Drake* provides at column 17, lines 25-59:

> The major functional areas that are addressed by the auditor/investigator GUI are as follows: manual raw audit file management, archive/restore; configure automated audit file archive; and maintenance of an audit file archive database; event detection system user identification and authentication; database connectivity with filter and sort capabilities for selecting and displaying event data in tabular format; an indicator on a tabular GUI display window to indicate when filtering is active; user interface for saving and selecting multiple filter and sort templates (setups) by user defined names (these saved setups are associated with the user, and are available when the user logs in to event detection system); allowing each authenticated user to save a default GUI setup, including the default filter and sort setup; an event status bar for graphical display of the highest event severity level, which has not yet been observed (after the initial observation of an event in the tabular display, the status bar will no longer use that event to update the status bar); user interface allowing the user to mark selected datasets in the database as "responded to" (events that have been responded to, will no longer be displayed in the default event display mode, but may be re-selected for display through a filter); print selected datasets; export selected datasets to a file; display selected datasets in chart form; display the status of distributed event detection system executables; display and print charts, including bar charts (n selected items horizontal, by value vertical, or n selected items horizontal, by m selected items deep, by value vertical from a currently filtered data set); creating chart templates; saving chart templates by name; creating a statistical template for a statistical viewer; saving statistical templates; displaying and printing statistical data, and statistical charts; viewing user status, including login status of user(s), vacation/tagged user status; and default data set filtering for selected users.

While *Drake* mentions use of templates in GUI 16, which retrieves normalized audit data from database 12, *Drake* provides no teaching or suggestion of a template for defining an output comprising raw audit data. Rather, *Drake* transforms its raw audit data to a normalized format, which is stored to database 12, and the GUI 16 accesses such normalized data rather than the raw audit data.

In response to the above arguments, the June 29th Office Action asserts (at pages 11-12 thereof):

> In 9:45-60, Drake teaches a process, which includes the collector 26 and the parser 20, where the audit acquisitions (audit data) are unique to the audit source, and each audit source being unique (i.e., desired format), and further stating that there is a different collector for each operating system application on a given format, Examiner interprets this to be the Equivalent function of Applicant's desired format defined by template, since different formats are able to be mapped to different formats, and Drake in 11, also

> discloses the use of Expert systems to convert data into different formats such as in the collection of statistical profiles, see 11:13-18.

However, as discussed above, while *Drake* teaches collecting raw audit data, such raw audit data itself is not output in a format defined by a template. Rather, *Drake* teaches that the raw audit data is converted to a normalized, standard format, and stored to database 12. The GUI 16 accesses such normalized data of database 12, rather than the raw audit data. While *Drake* provides no teaching of a template for defining the normalized format or any format of the individual audit sources, the Office Action disregards this deficiency of *Drake* and asserts that *Drake* anticipates claims 1, 14, and 37. It is impermissible to selectively ignore an element of the claims, such as the element reciting that a template defines a desired format for an output that comprises raw audit data. *Drake* provides no teaching of using a template for defining a format for converting raw audit data to normalized data. Even if such a template were used in *Drake* for this conversion process, such template would not define an output that comprises raw audit data, but would instead define the output for the converted normalized data, which is then stored to database 12. Again, any use a template for defining a desired format in *Drake* is used for outputting data from database 12, which is not raw audit data but is instead normalized audit data.

In view of the above, *Drake* fails to teach at least the above elements of independent claims 1, 14, and 37 and therefore these claims are not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claims 1, 14, and 37 be overturned.

Dependent claims 13 and 15-18 each depend either directly or indirectly from one of independent claims 1 and 14. Since Appellant believes that claims 1 and 14 are of patentable merit for the reasons discussed above, it follows *a fortiori* that dependent claims 13 and 15-18 must also be allowable because they carry with them all of the limitations of the claims from which they depend in addition to their own supplied limitations.

Independent Claim 54

Independent claim 54 recites in part "generating said output presentation that includes at least a portion of said raw audit data, wherein said output presentation comprises said desired format as defined by said audit transformation template" (emphasis added). As

discussed above with independent claims 1, 14, and 37, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data.

As discussed above, in response to this argument, the June 29th Office Action asserts (at pages 11-12 thereof) that *Drake* teaches unique audit sources which collect raw audit data that is then converted to a normalized format. While *Drake* provides no teaching of a template for defining the normalized format or any format of the individual audit sources, the Office Action disregards this deficiency of *Drake* and asserts that it anticipates claim 54. Claim 54 further specifies that the generated output is an "output presentation". Thus, even if *Drake* did teach use of a template for converting raw audit data to a normalized format (which it does not), *Drake* certainly fails to teach a template for defining the format of an "output presentation". The normalized format of *Drake* is not an output presentation, but is instead an internal format that is merely stored to database 12. Any output presentation in *Drake* is then generated by GUI 16 for presenting the normalized data of database 12, not raw audit data.

Thus, for these further reasons, *Drake* fails to teach all elements of independent claim 54, and therefore claim 54 is not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claim 54 be overturned.

Dependent Claim 10

Dependent claim 10 depends from independent claim 1, and thus inherits all limitations of independent claim 1. It is respectfully submitted that dependent claim 10 is allowable at least because of its dependency from independent claim 1 for the reasons discussed above.

Moreover, dependent claim 10 further recites "wherein said template comprises at least one conditional element." *Drake* fails to teach a template comprising a conditional element. The June 29th Office Action cites column 2, lines 45-55 of *Drake* as teaching such a template comprising a conditional element. Column 2, lines 40-56 of *Drake* provides:

> U.S. Pat. No. 5,557,742 (Smaha et al.), incorporated herein by reference, describes a method and system for detecting intrusion and misuse of data processing systems. The system uses processing system inputs, which

> include processing system audit trail records, system log file data, and system security state data information to detect and report processing system intrusions and misuses. A misuse selection mechanism allows the detection system to analyze the process inputs for a selected subset of misuses. The processing system inputs are then converted into states that are compared, through the misuse engine, to a predefined set of states and transitions until a selected misuse is detected. Once a misuse has been detected, an output mechanism generates a signal for use by a notification and storage mechanism. The detection system then generates a text-based output report for a user to view or store.

The above portion of *Drake* makes no mention of a template that defines an output that comprises raw audit data, and it certainly fails to teach such a template that comprises a conditional element. The above portion of *Drake* mentions that a "misuse selection mechanism" allows the detection system to analyze the process inputs for a selected subset of misuses. The misuses are then converted into states that are compared to a predefined set of states and transitions until a selected misuse is detected. While this mentions making comparisons of misuse states, it provides no teaching of a template that defines an output comprising raw audit data, wherein such template comprises at least one conditional element. The June 29th Office Action appears to be asserting that any comparison performed by a system satisfies the recited element of claim 10. This is improper. Claim 10 clearly recites that the "template comprises at least one conditional element", and *Drake* simply fails to provide any teaching of such a template that comprises a conditional element. To assert that the above-quoted portion of *Drake* teaches this element of claim 10 disregards the elements recited by claim 10.

In view of the above, *Drake* fails to teach the further element of claim 10, and therefore claim 10 is not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claim 10 be overturned.

Dependent Claim 11

Dependent claim 11 depends from claim 10, which depends independent claim 1, and thus claim 11 inherits all limitations of claims 1 and 10. It is respectfully submitted that dependent claim 11 is allowable at least because of its dependency from claims 1 and 10 for the reasons discussed above.

Moreover, dependent claim 11 further recites "wherein said at least one conditional element dictates that said output is to have a particular format if a condition is satisfied, otherwise said output is to have a different format." As discussed above with claim 10, *Drake* fails to teach a template comprising a conditional element. Further, *Drake* fails to teach such a conditional element that dictates that the output is to have a particular format if a condition is satisfied and otherwise have a different format.

The June 29th Office Action cites column 7, lines 25-31 of *Drake* as teaching this element of claim 11, *see* page 3 of the Office Action. Column 7, lines 25-31 of *Drake* provides:

> The process of converting audit data from its raw form into Virtual Records is referred to herein as audit parsing, or parsing. Audit parsing is broken down into several steps. Depending on the raw format of the event records, and the location of processing elements (there is a great deal of flexibility in locating processing elements), some steps may not be necessary in some variations and on some platforms.

The above portion of *Drake* makes no mention of a template that defines an output that comprises raw audit data, and it certainly fails to teach such a template that comprises a conditional element. Further, the above portion of *Drake* makes no mention of an output having a particular format if a condition is satisfied and otherwise having a different format. Rather, the above portion of *Drake* simply mentions that an audit parsing process is used to convert data from its raw form into Virtual Records, which are in normalized format. This does not teach a template that defines output that comprises raw audit data (*see* claim 1 from which claim 11 indirectly depends). First, the above portion of *Drake* does not teach use of a template, and certainly not a template that comprises a conditional element. Further, this portion of *Drake* fails to teach generating an output that comprises raw audit data, but instead teaches a conversion from the raw form into a normalized format.

In view of the above, *Drake* fails to teach the further element of claim 11, and therefore claim 11 is not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claim 11 be overturned.

Dependent Claims 40, 44, and 51

Dependent claim 40 depends from independent claim 1, and thus inherits all limitations of independent claim 1. It is respectfully submitted that dependent claim 40 is allowable at least because of its dependency from independent claim 1 for the reasons discussed above.

Dependent claim 44 depends from independent claim 14, and thus inherits all limitations of independent claim 14. It is respectfully submitted that dependent claim 44 is allowable at least because of its dependency from independent claim 14 for the reasons discussed above.

Dependent claim 51 depends from independent claim 37, and thus inherits all limitations of independent claim 37. It is respectfully submitted that dependent claim 51 is allowable at least because of its dependency from independent claim 37 for the reasons discussed above.

Moreover, dependent claim 40 further recites "wherein said generated output comprises presentation output." Similarly, dependent claim 44 further recites "wherein said code executable to generate output comprises: code executable to generate presentation output." Similarly, dependent claim 51 further recites "wherein said function executable to generate output comprises: function executable to generate output presentation."

Drake fails to teach these further elements of claims 40, 44, and 51. As discussed above with independent claims 1, 14, and 37, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data.

Also, as discussed above with independent claim 54, *Drake* fails to teach or suggest that the generated output comprises presentation output. *Drake* teaches unique audit sources which collect raw audit data that is then normalized and formatted for storage to a database. *Drake* teaches that its raw audit data is converted into a normalized format, i.e., Virtual Records. *Drake* provides no teaching of a template for defining the normalized format. Claim 40 further specifies that the generated output is a "presentation output". Similarly,

claim 44 further specifies code executable to generate presentation output. Also, claim 51 further specifies a function executable to generate output presentation. Thus, even if *Drake* did teach use of a template for converting raw audit data to a normalized format (which it does not), *Drake* certainly fails to teach a template for defining the format of an "output presentation". The normalized format of *Drake* (i.e., the Virtual Records) is not a presentation output, but is instead an internal format that is merely stored to database 12. Any presentation output in *Drake* is then generated by GUI 16 for presenting the normalized data of database 12, not raw audit data.

Thus, for these further reasons, *Drake* fails to teach all elements of claims 40, 44, and 51, and therefore claims 40, 44, and 51 are not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claims 40, 44, and 51 be overturned.

Dependent Claims 41, 45, and 52

Dependent claim 41 depends from claim 40, which depends from independent claim 1, and thus claim 41 inherits all limitations of claims 1 and 40. It is respectfully submitted that dependent claim 41 is allowable at least because of its dependency from claims 1 and 40 for the reasons discussed above.

Dependent claim 45 depends from claim 44, which depends from independent claim 14, and thus claim 45 inherits all limitations of claims 14 and 44. It is respectfully submitted that dependent claim 45 is allowable at least because of its dependency from claims 14 and 44 for the reasons discussed above.

Dependent claim 52 depends from claim 51, which depends from independent claim 37, and thus claim 52 inherits all limitations of claims 37 and 51. It is respectfully submitted that dependent claim 52 is allowable at least because of its dependency from claims 37 and 51 for the reasons discussed above.

Moreover, dependent claim 41 further recites "wherein said presentation output comprises at least one selected from the group consisting of: presentation output to a display, and presentation output to a printer." Similarly, dependent claim 45 recites "wherein said presentation output comprises at least one selected from the group consisting of: presentation

output to a display, and presentation output to a printer." Similarly, dependent claim 52 further recites "wherein said output presentation comprises at least one selected from the group consisting of: output presentation to a display, and output presentation to a printer."

As discussed above with claim 40, *Drake* fails to teach generating presentation output that comprises raw audit data in a format defined by a template. *Drake* further fails to teach such a presentation output that is output to a display or to a printer, as recited by claims 41, 45, and 52. Any presentation output that is generated by *Drake* (e.g., via GUI 16) does not comprise raw audit data, but instead outputs normalized audit data from database 12.

Thus, *Drake* fails to teach all elements of claims 41, 45, and 52, and therefore claims 41, 45, and 52 are not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claims 41, 45, and 52 be overturned.

Dependent Claims 42, 46, and 53

Dependent claim 42 depends from claim 40, which depends from independent claim 1, and thus claim 42 inherits all limitations of claims 1 and 40. It is respectfully submitted that dependent claim 42 is allowable at least because of its dependency from claims 1 and 40 for the reasons discussed above.

Dependent claim 46 depends from claim 44, which depends from independent claim 14, and thus claim 46 inherits all limitations of claims 14 and 44. It is respectfully submitted that dependent claim 46 is allowable at least because of its dependency from claims 14 and 44 for the reasons discussed above.

Dependent claim 53 depends from claim 52, which depends from claim 51, which depends from independent claim 37; and thus claim 53 inherits all limitations of claims 37, 51, and 52. It is respectfully submitted that dependent claim 53 is allowable at least because of its dependency from claims 37, 51, and 52 for the reasons discussed above.

Moreover, dependent claim 42 further recites "wherein said presentation output comprises at least one selected from the group consisting of: presentation output by a browser, presentation output by a spreadsheet program, and presentation output by an application program." Similarly, dependent claim 46 further recites "wherein said

presentation output comprises at least one selected from the group consisting of: presentation output by a browser, presentation output by a spreadsheet program, and presentation output by an application program." Similarly, dependent claim 53 further recites: "wherein said output presentation comprises at least one selected from the group consisting of: output presentation by a browser, output presentation by a spreadsheet program, and output presentation by an application program."

As discussed above with claim 40, *Drake* fails to teach generating presentation output that comprises raw audit data in a format defined by a template. *Drake* further fails to teach such a presentation output that is output by a browser, spreadsheet program, or application program, which comprises raw audit data, but instead outputs normalized audit data from database 12.

Thus, *Drake* fails to teach all elements of claims 42, 46, and 53, and therefore claims 42, 46, and 53 are not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claims 42, 46, and 53 be overturned.

Dependent Claims 43 and 47

Dependent claim 43 depends from independent claim 1, and thus inherits all limitations of independent claim 1. It is respectfully submitted that dependent claim 43 is allowable at least because of its dependency from independent claim 1 for the reasons discussed above.

Dependent claim 47 depends from independent claim 14, and thus inherits all limitations of independent claim 14. It is respectfully submitted that dependent claim 47 is allowable at least because of its dependency from independent claim 14 for the reasons discussed above.

Moreover, dependent claim 43 further recites a "user interface for receiving from a user input defining said template." Similarly, dependent claim 47 further recites "code executable to receive from a user input defining said audit transformation template."

Drake fails to teach such a user interface for receiving input defining a template. The June 29th Office Action (at pages 4-5 thereof) cites column 17, lines 25-40 of *Drake* as teaching the above element of claim 43. Column 17, lines 25-40 of *Drake* provides:

> The major functional areas that are addressed by the auditor/investigator GUI are as follows: manual raw audit file management, archive/restore; configure automated audit file archive; and maintenance of an audit file archive database; event detection system user identification and authentication; database connectivity with filter and sort capabilities for selecting and displaying event data in tabular format; an indicator on a tabular GUI display window to indicate when filtering is active; user interface for saving and selecting multiple filter and sort templates (setups) by user defined names (these saved setups are associated with the user, and are available when the user logs in to event detection system); allowing each authenticated user to save a default GUI setup, including the default filter and sort setup; an event status bar for graphical display of the highest event severity level, which has not yet been observed....

The above portion of *Drake* mentions that its auditor/investigator GUI provides a user interface for saving and selecting multiple filter and sort templates (setups) by user defined names. While this allows a user to save and select various "sort" templates, it does not teach that user input defines a template that defines the format for an output that comprises raw audit data.

Thus, *Drake* fails to teach all elements of claims 43 and 47, and therefore claims 43 and 47 are not anticipated by *Drake*. Accordingly, Appellant respectfully requests that the rejection of claims 43 and 47 be overturned.

II. Rejections Under 35 U.S.C. § 103(a) over *Drake* in view of *Sutton*

Claims 2-9, 19-26, and 38-39 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Drake* in view of *Sutton*. Dependent claims 2-9, 19-26, and 38-39 each depend either directly or indirectly from one of independent claims 1, 14, and 37, and thus inherit all limitations of the respective independent claim from which they depend. It is respectfully submitted that dependent claims 2-9, 19-26, and 38-39 are allowable not only because of their dependency from their respective independent claims for the reasons discussed above, but also in view of their novel claim features (which both narrow the scope of the particular claims and compel a broader interpretation of the respective base claim from which they depend).

Dependent Claims 4, 7-9, 20-23, and 38-39

Dependent claims 4, 7-9, 20-23, and 38-39 each depend either directly or indirectly from one of independent claims 1, 14, and 37. Appellant believes that claims 1, 14, and 37 are of patentable merit for the reasons discussed above. Further, the June 29th Office Action does not rely on *Sutton* to cure the deficiencies of *Drake* identified above for claims 1, 14, and 37, as *Sutton* does not cure such deficiencies. Thus, it follows *a fortiori* that because independent claims 1, 14, and 37 are of patentable merit, dependent claims 4, 7-9, 20-23, and 38-39 must also be allowable because they carry with them all of the limitations of the claims from which they depend in addition to their own supplied limitations.

Dependent Claim 2

Dependent claim 2 depends from independent claim 1, and thus inherits all limitations of independent claim 1. It is respectfully submitted that dependent claim 2 is allowable at least because of its dependency from independent claim 1 for the reasons discussed above, as *Sutton* does not cure the above-identified deficiencies of *Drake*.

Moreover, dependent claim 2 further recites "wherein said template comprises at least one constant element."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 2. The Office Action of June 29, 2005 concedes (at page 6 thereof) that *Drake* fails to teach or suggest such a template that comprises at least one constant element. However, the Office Action asserts that *Sutton* teaches such a template comprising at least one constant element. Appellant respectfully disagrees, as discussed below.

At column 3, lines 2-12, *Sutton* provides:

> Briefly, the invention relates to a hierarchical registration architecture for managing and organizing the collection of performance information, such as statistics and tracing, for an extensible operating system of a computer. The registration architecture, or registry, comprises a multi-linked tree data structure resident within a main memory for storing the performance information. Writer entities "register" their intent to collect and store

> performance information in the registry by creating objects, via novel API system calls, as nodes organized within the tree structure.

Thus, *Sutton* teaches a hierarchical registration architecture for managing and organizing performance information. *Sutton* further teaches at column 3, lines 31-38:

> According to another aspect of the invention, the novel registry provides a mechanism for extending the data types used for information collection. Preferably, the data types are abstract data types ranging from primitives (such as simple counters, histograms, maximum values, minimum values, averages and rates) to user-defined data types consisting of either entirely new data types or combinations of existing primitive data types.

Thus, *Sutton* teaches that various data types may be used for collection of the performance information, such as counters, averages, etc. This fails to teach or suggest a template comprising at least one constant element. Rather, the data types in *Sutton* contain variable information, such as counters, averages, etc., for the respective collected performance data at a given time. Further still, *Sutton* does not teach or suggest that the data types are part of a template, and particularly not part of a template that defines a desired format for generated output comprising raw audit data (*see* claim 1 from which claim 2 depends). *Sutton* further provides at column 9, lines 28-40 (a portion of which is cited by the June 29th Office Action as teaching the above element of claim 2):

> In accordance with another aspect of the invention, the classes of the novel registration architecture described above provide mechanisms for extending the data types used for information collection. Preferably, the data types are abstract data types ranging from primitives (such as simple counters, maximum values, minimum values, averages and rates) to user-defined data types consisting of either entirely new data types or combinations of existing primitive data types. An example of a single primitive data type involves the creation and use of a counter to record the number of requests issued to a disk. As a disk driver receives these requests, it simply increments the counter to record the number of requests received.

Thus, *Sutton* teaches that various data types can be used for collecting performance information, such as a counter data type for recording the number of requests issued to a disk. This in no way teaches or suggests a template comprising at least one constant element, as recited by claim 2. Rather, this teaches data types for collecting variable performance information, such as counters, averages, etc. Accordingly, such data types contain variable elements, instead of constant elements, as the information contained in a data type (e.g., a

counter) will vary over time as the corresponding performance to which the information relates varies.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 2, and therefore Appellant respectfully requests that the rejection of claim 2 under 35 U.S.C. §103 be overturned.

ii. Lack of Motivation to Combine Drake and Sutton

It is well settled that the fact that references can be combined or modified is not sufficient to establish a prima facie case of obviousness, *see* M.P.E.P. § 2143.01. The mere fact that references can be combined or modified does not render the resultant combination or modification obvious unless the prior art also suggests the desirability of the combination or modification. *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990), as cited in M.P.E.P. § 2143.01. The June 29th Office Action contends, at page 6 thereof, that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Drake and Sutton because, using constant elements ensures more reusability of templates."

However, neither *Drake* nor *Sutton* teaches or suggests using constant elements. Further, neither *Drake* nor *Sutton* teaches or suggests that the use of constant elements ensures more reusability of templates. Further still, neither *Drake* nor *Sutton* teaches or suggests that more reusability of templates is desirable. Furthermore, Appellant fails to understand how the use of constant elements promotes reusability of templates, as asserted by the June 29th Office Action, particularly when reporting varying information such as the performance information of *Sutton*. For example, if the performance information collected at a first point in time in *Sutton* is reported as a constant element, a separate template would be required for reporting different performance information at another point in time. Thus, contrary to the assertion of the June 29th Office Action, the use of constant elements for reporting information, such as the information with which *Sutton* is concerned, does not improve reusability of templates.

In view of the above, the motivation provided by the Examiner is improper, as the applied references must establish the desirability for making the modification. Accordingly,

for this further reason, Appellant respectfully requests that the rejection of claim 2 under 35 U.S.C. §103 be overturned.

Dependent Claims 3 and 19

Dependent claim 3 depends from claim 2, which depends from independent claim 1, and thus claim 3 inherits all limitations of claims 1-2. It is respectfully submitted that dependent claim 3 is allowable at least because of its dependency from claims 1-2 for the reasons discussed above.

Dependent claim 19 depends from independent claim 14, and thus inherits all limitations of claim 14. It is respectfully submitted that dependent claim 19 is allowable at least because of its dependency from independent claim 14 for the reasons discussed above.

Moreover, dependent claim 3 further recites "wherein said at least one constant element is included verbatim in said output." Similarly, dependent claim 19 further recites "wherein said audit transformation template comprises at least one constant element that is included verbatim in said output."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claims 3 and 19. The Office Action of June 29, 2005 concedes (at page 6 thereof) that *Drake* fails to teach or suggest a template that comprises at least one constant element. However, the Office Action cites *Drake* (*see* page 7 of the Office Action) as teaching that the at least one constant element is included verbatim in an output. Since the Examiner concedes that *Drake* does not teach or suggest a template comprising at least one constant element, Appellant fails to understand how *Drake* could teach that such a constant element is included verbatim in an output. The June 29th Office Action cites to column 4, lines 5-10 of *Drake* as teaching this element of claim 3. Column 4, lines 2-12 of *Drake* provides:

> In a further variation of the system of the one embodiment, the detector is coupled to an output of the parser, and the detector detects audit events in response to the Virtual Records generated by the parser and generates Virtual Records of the second type in response thereto.
>
> In yet another further variation of the system of the one embodiment,

> an input of the detector is coupled to an output of the database, and the detector detects audit events in response to the Virtual Records in the database, and generates Virtual Records of the second type in response thereto. Also, an output of the detector may be coupled to the inserter, so that Virtual Records of the second type generated by the detector are inserted into the database by the inserter.

As can be seen, nothing in this cited portion of *Drake* teaches or suggests a template comprising at least one constant element, wherein the constant element is included verbatim in an output. As the June 29th Office Action acknowledges, *Drake* simply fails to teach a template comprising at least one constant element, and thus *Drake* further fails to teach or suggest that such a constant element is included verbatim in an output.

Further, the June 29th Office Action does not rely upon *Sutton* as teaching this further element of claims 3 and 19, as *Sutton* also fails to teach or suggest this element. As described above with claim 2, *Sutton* does not teach or suggest a template that comprises at least one constant element, and thus *Sutton* also fails to teach or suggest this further element of claims 3 and 19.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claims 3 and 19, and therefore Appellant respectfully requests that the rejection of claims 3 and 19 under 35 U.S.C. §103 be overturned.

ii. Lack of Motivation to Combine Drake and Sutton

In its rejection of claim 3, the June 29th Office Action does not rely upon *Sutton* for the further element of claim 3. Similarly, in rejecting claim 19, the June 29th Office Action merely references the rationale provided for claim 3. Thus, the rejection of claims 3 and 19 provide no motivation for combining *Drake* and *Sutton* beyond that provided for claim 2. As discussed above with claim 2, the motivation provided by the Examiner is improper, as the applied references must establish the desirability for making the modification. Accordingly, for this further reason, Appellant respectfully requests that the rejection of claims 3 and 19 under 35 U.S.C. §103 be overturned.

Dependent Claims 5 and 24

Dependent claim 5 depends from claim 4, which depends from independent claim 1, and thus claim 5 inherits all limitations of claims 1 and 4. It is respectfully submitted that dependent claim 5 is allowable at least because of its dependency from independent claim 1 for the reasons discussed above.

Dependent claim 24 depends from claim 21, which depends from claim 20, which depends from independent claim 14; and thus claim 24 inherits all limitations of claims 14, 20, and 21. It is respectfully submitted that dependent claim 24 is allowable at least because of its dependency from independent claim 14 for the reasons discussed above.

Moreover, dependent claim 4 further recites "wherein said template comprises at least one variable element", and dependent claim 5 further recites "wherein said at least one variable element identifies a particular portion of the raw audit data to be included in said output." Similarly, dependent claim 24 further recites "wherein said at least one variable element each identify a particular type of audit information to be included in said output."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claims 5 and 24. The Office Action of June 29, 2005 concedes (at page 7 thereof) that *Drake* fails to teach or suggest a template that comprises at least one variable element. However, the Office Action cites *Drake* (*see* page 7 of the Office Action) as teaching that the at least one variable element identifies a particular portion of the raw audit data to be included in an output. Since the Examiner concedes that *Drake* does not teach or suggest a template comprising at least one variable element, Appellant fails to understand how *Drake* could teach that such a variable element identifies a particular portion of raw audit data to be included in an output. The June 29th Office Action cites to column 4, lines 3-25 of *Drake* as teaching this element of claim 5. Column 4, lines 2-27 of *Drake* provides:

> In a further variation of the system of the one embodiment, the detector is coupled to an output of the parser, and the detector detects audit events in response to the Virtual Records generated by the parser and generates Virtual Records of the second type in response thereto.
>
> In yet another further variation of the system of the one embodiment, an input of the detector is coupled to an output of the database, and the detector detects audit events in response to the Virtual Records in the database, and generates Virtual Records of the second type in response

> thereto. Also, an output of the detector may be coupled to the inserter, so that Virtual Records of the second type generated by the detector are inserted into the database by the inserter.
>
> In another further variation, an output of said detector is coupled to the inserter, and Virtual Records of the second type generated by the detector are inserted into the database by the inserter.
>
> In an additional variation, the event detection system also includes a user interface coupled to the database. The user interface filters Virtual Records in the database based on a filter criteria.
>
> In variations, the detector may perform rule-based analysis of the Virtual Records generated by the parser, and/or may perform statistical analysis of the Virtual Records generated by the parser.

As can be seen, nothing in this cited portion of *Drake* teaches or suggests a template comprising at least one variable element, wherein the variable element identifies a particular portion of raw audit data to be included in an output. As the June 29th Office Action acknowledges, *Drake* simply fails to teach a template comprising at least one variable element, and thus *Drake* further fails to teach or suggest that such a variable element identifies a particular portion of raw audit data to be included in an output. Additionally, as discussed above with claim 1, *Drake* does not teach including raw audit data in an output, and thus *Drake* fails to teach a variable element that identifies a portion of such raw audit data to include in an output.

Further, the June 29th Office Action does not rely upon *Sutton* as teaching this further element of claim 5 and 24, as *Sutton* also fails to teach or suggest this element.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claims 5 and 24, and therefore Appellant respectfully requests that the rejection of claims 5 and 24 under 35 U.S.C. §103 be overturned.

ii. Lack of Motivation to Combine Drake and Sutton

In its rejection of claim 5, the June 29th Office Action does not rely upon *Sutton* for the further element of claim 5. Thus, the rejection of claim 5 provides no motivation for combining *Drake* and *Sutton* beyond that provided for claim 4. In addressing claim 4, the June 29th Office Action asserts that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Drake and Sutton because, using

variable elements would make the templates more customizable." Page 7 of the Office Action.

However, neither *Drake* nor *Sutton* teaches or suggests that the use of variable elements makes templates more customizable. Further, neither *Drake* nor *Sutton* teaches or suggests that making templates more customizable is desirable. Furthermore, the language of the recited motivation is circular in nature, stating that it is obvious to make the modification because it is obvious to achieve the result. That is, the language merely asserts that it would be obvious to include at least one variable element in a template to achieve the benefits (e.g., customization) of including such a variable element in a template. This does not identify any motivation that the applied references provide, but merely asserts with the benefit of hindsight that such a modification is obvious. The mere fact that references can be combined or modified does not render the resultant combination or modification obvious unless the prior art also suggests the desirability of the combination or modification. *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990), as cited in M.P.E.P. § 2143.01. Thus, the motivation provided by the Examiner is improper, as the cited prior art reference must establish the desirability for making the modification.

Accordingly, for this further reason, Appellant respectfully requests that the rejection of claims 5 and 24 under 35 U.S.C. §103 be overturned.

Dependent Claim 6

Dependent claim 6 depends from claim 5, which depends from 4, which depends from independent claim 1, and thus claim 6 inherits all limitations of claims 1 and 4-5. It is respectfully submitted that dependent claim 6 is allowable at least because of its dependency from independent claim 1 and dependent claim 5 for the reasons discussed above.

Moreover, dependent claim 6 further recites "wherein said at least one variable element identifies a location within said output at which said particular portion of the raw audit data is to be arranged."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 6. The Office Action of June 29, 2005 concedes (at page 7 thereof) that *Drake* fails to teach or suggest a template that comprises at least one variable element. However, the Office Action cites *Drake* (*see* page 7 of the Office Action) as teaching that the at least one variable element identifies a location within an output at which a particular portion of the raw audit data is to be arranged. Since the Examiner concedes that *Drake* does not teach or suggest a template comprising at least one variable element, Appellant fails to understand how *Drake* could teach that such a variable element identifies a location within an output at which a particular portion of the raw audit data is to be arranged.

The June 29th Office Action cites to column 4, lines 3-25 of *Drake* as teaching this element of claim 6. However, neither this portion, nor any other portion of *Drake*, teaches or suggests use of a variable element in a template for identifying a location within an output at which a particular portion of the raw audit data is to be arranged. As the June 29th Office Action acknowledges, *Drake* simply fails to teach a template comprising at least one variable element, and thus *Drake* further fails to teach or suggest that such a variable element identifies a location within an output at which a particular portion of the raw audit data is to be arranged. Additionally, as discussed above with claim 1, *Drake* does not teach including raw audit data in an output, and thus *Drake* fails to teach a variable element that identifies a location within an output at which a particular portion of the raw audit data is to be arranged.

Further, the June 29th Office Action does not rely upon *Sutton* as teaching this further element of claim 6, as *Sutton* also fails to teach or suggest this element. That is, *Sutton* does not teach or suggest a template that comprises at least one variable element that identifies a location within an output at which a particular portion of raw audit data is to be arranged.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 6, and therefore Appellant respectfully requests that the rejection of claim 6 under 35 U.S.C. §103 be overturned.

ii. Lack of Motivation to Combine Drake and Sutton

In its rejection of claim 6, the June 29th Office Action does not rely upon *Sutton* for the further element of claim 6. Thus, the rejection of claim 6 provides no motivation for

combining *Drake* and *Sutton* beyond that provided for claim 4. As discussed above with claim 5, the motivation provided by the Examiner in addressing claim 4 is improper, as the applied references must establish the desirability for making the modification. Accordingly, for this further reason, Appellant respectfully requests that the rejection of claim 6 under 35 U.S.C. §103 be overturned.

Dependent claim 25

While the June 29th Office Action asserts that claim 25 is rejected under 35 U.S.C. § 103(a) as being obvious over *Drake* in view of *Sutton*, in its treatment of claim 25 the Office Action merely refers to the rationale provided for claim 10, *see* page 8 of the Office Action. As discussed above, claim 10 is rejected under 35 U.S.C. § 102 as being anticipated by *Drake*. Thus, for the reasons discussed above with claim 10, Appellant respectfully submits that *Drake* fails to teach all elements of claim 25. Further, a prima facie case of obviousness has not been established by the Office Action for claim 25 because the Office Action merely refers to the 35 U.S.C. § 102 of claim 10. Thus, the rejection fails to identify any element that is lacking from *Drake*, any teaching or suggestion of the lacking element in *Sutton*, and any motivation for combining the teaching of *Sutton* with *Drake* to supply the lacking element. *Sutton* also fails to teach or suggest a template with a conditional element, as recited by claim 25.

Accordingly, Appellant respectfully requests that the rejection of claim 25 should be overturned as improper.

Independent Claim 26 and Dependent Claims 27, 30-32, and 34-36

Appellant respectfully submits that independent claim 26 is not obvious under 35 U.S.C. § 103(a) over *Drake* in view of *Sutton*, as discussed further below. To establish a prima facie case of obviousness, three basic criteria must be met. *See* M.P.E.P. § 2143. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success.

Finally, the prior art references must teach or suggest all the claim limitations. Without conceding any other criteria, Appellant respectfully asserts that the rejection does not satisfy the third criteria.

Independent claim 26 recites in part "generating an output that includes at least a portion of said raw audit data, wherein said output comprises said desired format as defined by said audit transformation template" (emphasis added). As described above, *Drake* fails to teach or suggest an audit transformation template that defines a desired format for generated output that includes at least a portion of raw audit data. As discussed above with independent claims 1, 14, 37, and 54, *Drake* provides no teaching or suggestion of a template for defining an output comprising raw audit data. Rather, *Drake* transforms its raw audit data to a normalized format, which is stored to database 12, and the GUI 16 accesses such normalized data rather than the raw audit data.

Sutton also fails to teach or suggest this element of claim 26.

Accordingly, the applied combination of *Drake* and *Sutton* fails to teach or suggest at least the above element of independent claim 26.

In view of the above, independent claim 26 is not obvious under 35 U.S.C. § 103(a) over *Drake* in view of *Sutton*. Therefore, Appellant respectfully requests that this rejection be overturned.

Dependent claims 27, 30-32, and 34-36 each depend either directly or indirectly from independent claim 26. Since Appellant believes that claim 26 is of patentable merit for the reasons discussed above, it follows *a fortiori* that dependent claims 27, 30-32, and 34-36 must also be allowable because they carry with them all of the limitations of the claims from which they depend in addition to their own supplied limitations.

Dependent Claim 28

Dependent claim 28 depends from independent claim 26, and thus inherits all limitations of independent claim 26. It is respectfully submitted that dependent claim 28 is allowable at least because of its dependency from independent claim 26 for the reasons discussed above.

Moreover, dependent claim 28 further recites "receiving input from a user for creating said audit transformation template."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 28. The Office Action of June 29, 2005 asserts (at page 9 thereof) that *Drake* teaches this element of claim 28, citing column 16, lines 1-7 of *Drake*. Column 16, lines 1-7 of *Drake* provides:

> The event detection system statistical process operates by tabulating events by categories resolved by user, platform, and interval. After a defined collection period for statistical profiling (typically three months), the statistical processor starts comparing short term category counts with the longer term profiles for deviations from a statistical profile, typically using mean and standard deviation calculations.

This portion of *Drake* merely teaches that statistical profiling can be performed for events tabulated by category. This in no way teaches or suggests receiving input from a user for creating an audit transformation template, as recited by claim 28.

Further, the June 29th Office Action does not rely upon *Sutton* as teaching this further element of claim 28, as *Sutton* also fails to teach or suggest this element.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 28, and therefore Appellant respectfully requests that the rejection of claim 28 under 35 U.S.C. §103 be overturned.

Dependent Claim 29

Dependent claim 29 depends from independent claim 26, and thus inherits all limitations of independent claim 26. It is respectfully submitted that dependent claim 29 is allowable at least because of its dependency from independent claim 26 for the reasons discussed above.

Moreover, dependent claim 29 further recites "wherein said audit transformation template comprises at least one constant element that is included verbatim in said output."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 29. The Office Action of June 29, 2005 concedes (at page 6 thereof) that *Drake* fails to teach or suggest a template that comprises at least one constant element. However, the Office Action cites *Drake* (*see* page 7 of the Office Action) as teaching that the at least one constant element is included verbatim in an output. Since the Examiner concedes that *Drake* does not teach or suggest a template comprising at least one constant element, Appellant fails to understand how *Drake* could teach that such a constant element is included verbatim in an output. The June 29th Office Action cites to its rationale for claim 3 in rejecting claim 29, and in rejecting claim 3 the Office Action cites to column 4, lines 5-10 of *Drake* as teaching this element. Column 4, lines 2-12 of *Drake* provides:

> In a further variation of the system of the one embodiment, the detector is coupled to an output of the parser, and the detector detects audit events in response to the Virtual Records generated by the parser and generates Virtual Records of the second type in response thereto.
>
> In yet another further variation of the system of the one embodiment, an input of the detector is coupled to an output of the database, and the detector detects audit events in response to the Virtual Records in the database, and generates Virtual Records of the second type in response thereto. Also, an output of the detector may be coupled to the inserter, so that Virtual Records of the second type generated by the detector are inserted into the database by the inserter.

As can be seen, nothing in this cited portion of *Drake* teaches or suggests a template comprising at least one constant element, wherein the constant element is included verbatim in an output. As the June 29th Office Action acknowledges, *Drake* simply fails to teach a template comprising at least one constant element, and thus *Drake* further fails to teach or suggest that such a constant element is included verbatim in an output.

Further, the June 29th Office Action does not rely upon *Sutton* as teaching this further element of claim 29, as *Sutton* also fails to teach or suggest this element. As described above with claim 2, *Sutton* does not teach or suggest a template that comprises at least one constant element, and thus *Sutton* also fails to teach or suggest this further element of claim 29.

In view of the above, the combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 29, and therefore Appellant respectfully requests that the rejection of claim 29 under 35 U.S.C. §103 be overturned.

ii. Lack of Motivation to Combine Drake and Sutton

In rejecting claim 29, the June 29th Office Action merely cites to the rationale for rejecting claim 3. In its rejection of claim 3, the June 29th Office Action does not rely upon *Sutton* for the further element of claim 3. Thus, the rejection of claim 29 provide no motivation for combining *Drake* and *Sutton* beyond that provided for claim 2. As discussed above with claim 2, the motivation provided by the Examiner is improper, as the applied references must establish the desirability for making the modification. Accordingly, for this further reason, Appellant respectfully requests that the rejection of claim 29 under 35 U.S.C. §103 be overturned.

Dependent Claim 33

Dependent claim 33 depends from independent claim 26, and thus inherits all limitations of independent claim 26. It is respectfully submitted that dependent claim 33 is allowable at least because of its dependency from independent claim 26 for the reasons discussed above.

Moreover, dependent claim 33 further recites "presenting said output to a user."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

Drake fails to teach this further element of claim 33. As discussed above with independent claim 26, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data.

Further, *Drake* fails to teach or suggest presenting such output (that comprises raw audit data) to a user. Rather, in *Drake*, any presentation to a user (having a format defined by a template) does not comprise raw audit data, but instead presents normalized audit data from database 12.

Sutton is not relied upon as teaching or suggesting this element, as it also fails to teach or suggest presenting such output to a user.

Thus, for these further reasons, the combination of *Drake* and *Sutton* fails to teach or suggest all elements of claim 33, and therefore claim 33 is not obvious in view of *Drake* and *Sutton*. Accordingly, Appellant respectfully requests that the rejection of claim 33 be overturned.

Dependent Claim 48

Dependent claim 48 depends from independent claim 26, and thus inherits all limitations of independent claim 26. It is respectfully submitted that dependent claim 48 is allowable at least because of its dependency from independent claim 26 for the reasons discussed above.

Moreover, dependent claim 48 further recites "wherein said generating an output comprises: generating an output presentation."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 48. As discussed above with independent claim 26, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data. Further, *Drake* fails to teach or suggest generating an output presentation (as discussed above with independent claim 54). Rather, in *Drake*, any output presentation (having a format defined by a template) does not comprise raw audit data, but instead presents normalized audit data from database 12.

Sutton is not relied upon as teaching or suggesting this element, as it also fails to teach or suggest generating an output presentation that comprises raw audit data.

Thus, for these further reasons, the combination of *Drake* and *Sutton* fails to teach or suggest all elements of claim 48, and therefore claim 48 is not obvious in view of *Drake* and *Sutton*. Accordingly, Appellant respectfully requests that the rejection of claim 48 be overturned.

Dependent Claim 49

Dependent claim 49 depends from claim 48, which depends from independent claim 26; and thus claim 49 inherits all limitations of claims 26 and 48. It is respectfully submitted that dependent claim 49 is allowable at least because of its dependency from independent claims 26 and 48 for the reasons discussed above.

Moreover, dependent claim 49 further recites "wherein said output presentation comprises at least one selected from the group consisting of: output presentation to a display, and output presentation to a printer."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 49. As discussed above with independent claim 26, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data. Further, *Drake* fails to teach or suggest generating an output presentation (as discussed above with independent claim 54 and dependent claim 48). *Drake* further fails to teach or suggest any such output presentation to a display or to a printer. Rather, in *Drake*, any output presentation (having a format defined by a template) to a display or a printer does not comprise raw audit data, but instead presents normalized audit data from database 12.

Sutton is not relied upon as teaching or suggesting this element, as it also fails to teach or suggest such element.

Thus, for these further reasons, the combination of *Drake* and *Sutton* fails to teach or suggest all elements of claim 49, and therefore claim 49 is not obvious in view of *Drake* and *Sutton*. Accordingly, Appellant respectfully requests that the rejection of claim 49 be overturned.

Dependent Claim 50

Dependent claim 50 depends from claim 49, which depends from claim 48, which depends from independent claim 26; and thus claim 50 inherits all limitations of claims 26,

48, and 49. It is respectfully submitted that dependent claim 50 is allowable at least because of its dependency from independent claims 26, 48, and 49 for the reasons discussed above.

Moreover, dependent claim 50 further recites "wherein said output presentation comprises at least one selected from the group consisting of: output presentation by a browser, output presentation by a spreadsheet program, and output presentation by an application program."

i. The Applied Combination Fails to Teach or Suggest All Claim Elements

The combination of *Drake* and *Sutton* fails to teach or suggest this further element of claim 50. As discussed above with independent claim 26, *Drake* does not teach or suggest a template that defines a desired format for an output that comprises raw audit data. Further, *Drake* fails to teach or suggest generating an output presentation (as discussed above with independent claim 54 and dependent claim 48). *Drake* further fails to teach or suggest any such output presentation to a display or to a printer (as discussed above with claim 49). Further still, *Drake* fails to teach or suggest that such output presentation is generated by a browser, spreadsheet program, or application program. Again, *Drake* fails to teach or suggest such an output presentation that comprises raw audit data.

Sutton is not relied upon as teaching or suggesting this element, as it also fails to teach or suggest such element.

Thus, for these further reasons, the combination of *Drake* and *Sutton* fails to teach or suggest all elements of claim 50, and therefore claim 50 is not obvious in view of *Drake* and *Sutton*. Accordingly, Appellant respectfully requests that the rejection of claim 50 be overturned.

III. Rejection Under 35 U.S.C. § 103(a) over *Drake* in view of *Maloney*

Claim 12

Also, claim 12 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Drake* in view of *Maloney*. Claim 12 depends from independent claim 1, and thus inherits all limitations of claim 1. Appellant respectfully submits that claim 12 is allowable at least for

the reasons discussed above with claim 1 because *Maloney* does not cure the deficiencies of *Drake* identified above for claim 1. Thus, the rejection of claim 12 should be overturned.

VIII. CLAIMS

A copy of the claims involved in the present appeal is attached hereto as Appendix A. As indicated above, the claims in Appendix A do include the amendments filed by Applicant on April 6, 2005.

IX. EVIDENCE

As noted in Appendix B hereto, no evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS

As noted in Appendix C hereto, no related proceedings are referenced in II. above, or copies of decisions in related proceedings are not provided.

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV568260183US in an envelope addressed to: Appeal Brief - Patents, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: 11-28-05

Typed Name: Joy H. Perigo

Signature: Joy H. Perigo

Respectfully submitted,

By: _____

Jody C. Bishop
Attorney/Agent for Applicant(s)
Reg. No. 44,034
Date: November 28, 2005
Telephone No. (214) 855-8007

APPENDIX A

Claims Involved in the Appeal of Application Serial No. 09/896,351

1. A system comprising:

operating system providing at least one routine capable of being invoked, and said operating system operable to collect raw audit data for invoked operating system routines;

data storage having said raw audit data stored thereto; and

software code executable by at least one processor to receive said raw audit data and generate output comprising at least a portion of said raw audit data in a desired format defined by a template.

2. The system of claim 1 wherein said template comprises at least one constant element.

3. The system of claim 2 wherein said at least one constant element is included verbatim in said output.

4. The system of claim 1 wherein said template comprises at least one variable element.

5. The system of claim 4 wherein said at least one variable element identifies a particular portion of the raw audit data to be included in said output.

6. The system of claim 5 wherein said at least one variable element identifies a location within said output at which said particular portion of the raw audit data is to be arranged.

7. The system of claim 1 wherein said raw audit data comprises a record for each invocation of an operating system routine, and wherein each record includes at least one type of audit information relating to execution of said invoked operating system routine.

8. The system of claim 7 wherein said at least one type of audit information includes at least one type selected from the group consisting of:

user identification, group identification, supplementary group identification, process identification, event identification, event count, event type, date, time, thread identification, system call, capabilities used, object, and result.

9. The system of claim 7 wherein said template comprises at least one variable element that each identifies a particular type of audit information to be included in said output.

10. The system of claim 1 wherein said template comprises at least one conditional element.

11. The system of claim 10 wherein said at least one conditional element dictates that said output is to have a particular format if a condition is satisfied, otherwise said output is to have a different format.

12. The system of claim 1 wherein said template defines a format selected from the group consisting of:

plain text, markup language, and comma separated format.

13. The system of claim 1 wherein said operating system comprises a kernel-level audit device driver for collecting said audit data.

14. A computer program product for generating audit data in a desired format, said audit data relating to execution of a routine, said computer program product comprising a computer-readable storage medium having computer-readable program code embodied in said medium, said computer-readable program code comprising:

code executable to access raw audit data stored in a data storage device, wherein said raw audit data comprises information relating to execution of at least one invoked routine;

code executable to access an audit transformation template; and

code executable to generate output comprising at least a portion of said raw audit data, said output having a format defined by said audit transformation template.

15. The computer program product of claim 14 wherein said raw audit data is collected by an operating system.

16. The computer program product of claim 14 wherein said at least one routine includes at least one invoked operating system routine.

17. The computer program product of claim 16 wherein said at least one invoked operating system routine is invoked by an application via system call.

18. The computer program product of claim 16 wherein said at least one invoked operating system routine is invoked via user command.

19. The computer program product of claim 14 wherein said audit transformation template comprises at least one constant element that is included verbatim in said output.

20. The computer program product of claim 14 wherein said template comprises at least one variable element.

21. The computer program product of claim 20 wherein said raw audit data comprises a record for each invocation of an operating system routine that is included within said raw audit data, and wherein each record includes at least one type of audit information relating to execution of an invoked operating system routine.

22. The computer program product of claim 21 wherein said at least one type of audit information includes at least one type selected from the group consisting of:

user identification, group identification, supplementary group identification, process identification, event identification, event count, event type, date, time, thread identification, system call, capabilities used, object, and result.

23. The computer program product of claim 22 wherein said raw audit data comprises multiple ones of said record, further comprising:

code executable to sort at least a portion of the multiple records based on at least one of said types of audit information.

24. The computer program product of claim 21 wherein said at least one variable element each identify a particular type of audit information to be included in said output.

25. The computer program product of claim 14 wherein said template comprises at least one conditional element, and wherein said conditional element dictates that said output is to have a first format if a condition is satisfied and have a different format if said condition is not satisfied.

26. A method of generating an output that includes audit data therein and has a desired format, said method comprising the steps of:

collecting raw audit data relating to the execution of one or more invoked routines;

storing said raw audit data to a data storage device;

accessing said raw audit data;

accessing an audit transformation template that defines a desired format; and

generating an output that includes at least a portion of said raw audit data, wherein said output comprises said desired format as defined by said audit transformation template.

27. The method of claim 26 wherein said raw audit data comprises information about at least one invoked operating system routine.

28. The method of claim 26 further comprising:

receiving input from a user for creating said audit transformation template.

29. The method of claim 26 wherein said audit transformation template comprises at least one constant element that is included verbatim in said output.

30. The method of claim 26 wherein said audit transformation template comprises at least one variable element.

31. The method of claim 30 wherein said at least one variable element identifies a particular type of audit information from said raw audit data to be included in said output.

32. The method of claim 31 wherein said particular type of audit information includes at least one type selected from the group consisting of:

user identification, group identification, supplementary group identification, process identification, event identification, event count, event type, date, time, thread identification, system call, capabilities used, object, and result.

33. The method of claim 26 further comprising the step of:

presenting said output to a user.

34. The method of claim 26 further comprising the step of:

storing said output to a file.

35. The method of claim 26 further comprising the step of:

inputting said output to an application for processing by said application.

36. The method of claim 26 further comprising the step of:

sorting said raw audit data based at least in part on at least one type of audit information included therein.

37. A library of software functions stored to a computer-readable medium comprising:

function executable to access raw audit data collected by an auditing program, wherein said raw audit data comprises information about at least one invoked routine of said operating system;

function executable to access a template defining an output format; and

function executable to generate output comprising at least a portion of said ~~collected~~ raw audit data, wherein said output has a format defined by said template.

38. The library of claim 37 wherein said function executable to access raw audit data, said function executable to access a template, and said function executable to generate output are distinct functions.

39. The library of claim 37 wherein said function executable to access raw audit data, said function executable to access a template, and said function executable to generate output are included within a common function.

40. The system of claim 1 wherein said generated output comprises presentation output.

41. The system of claim 40 wherein said presentation output comprises at least one selected from the group consisting of:

presentation output to a display, and presentation output to a printer.

42. The system of claim 40 wherein said presentation output comprises at least one selected from the group consisting of:

presentation output by a browser, presentation output by a spreadsheet program, and presentation output by an application program.

43. The system of claim 1 further comprising:

user interface for receiving from a user input defining said template.

44. The computer program product of claim 14 wherein said code executable to generate output comprises:

code executable to generate presentation output.

45. The computer program product of claim 44 wherein said presentation output comprises at least one selected from the group consisting of:

presentation output to a display, and presentation output to a printer.

46. The computer program product of claim 44 wherein said presentation output comprises at least one selected from the group consisting of:

presentation output by a browser, presentation output by a spreadsheet program, and presentation output by an application program.

47. The computer program product of claim 14 further comprising:

code executable to receive from a user input defining said audit transformation template.

48. The method of claim 26 wherein said generating an output comprises:

generating an output presentation.

49. The method of claim 48 wherein said output presentation comprises at least one selected from the group consisting of:

output presentation to a display, and output presentation to a printer.

50. The method of claim 49 wherein said output presentation comprises at least one selected from the group consisting of:

output presentation by a browser, output presentation by a spreadsheet program, and output presentation by an application program.

51. The library of claim 37 wherein said function executable to generate output comprises:

function executable to generate output presentation.

52. The library of claim 51 wherein said output presentation comprises at least one selected from the group consisting of:

output presentation to a display, and output presentation to a printer.

53. The library of claim 52 wherein said output presentation comprises at least one selected from the group consisting of:

output presentation by a browser, output presentation by a spreadsheet program, and output presentation by an application program.

54. A method of generating an output presentation that includes audit data therein and has a desired format, said method comprising the steps of:

receiving input defining an audit transformation template that defines a desired format for said output presentation;

collecting raw audit data relating to the execution of one or more invoked routines;

storing said raw audit data to a data storage device;

accessing said raw audit data;

accessing said audit transformation template that defines a desired format; and

generating said output presentation that includes at least a portion of said raw audit data, wherein said output presentation comprises said desired format as defined by said audit transformation template.

APPENDIX B

Evidence

None.

APPENDIX C

Related Proceedings

None.